CSC2021 Advanced Programming using Android and C++
CSC2022 Games Programming using Android and C++

Project Assessment Criteria

This document will provide you with really important information on how your project will be

Assessment

Final project assessment

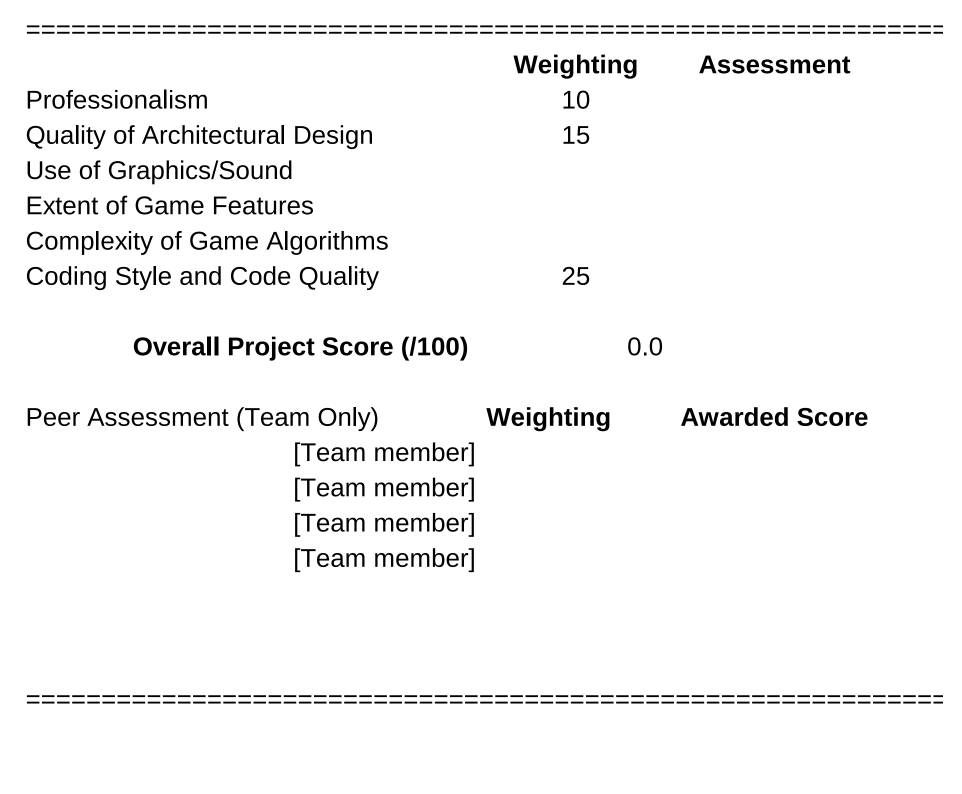
This section will record the final assessment of the developed game and will hence determine the awarded mark for the project component of the module. As a reminder, the project component is worth 1.5 module credits.

Important: The information in this section should be read early in the module and consulted frequently. It is very important to me that you understand how your project will be assessed as it will help you plan and develop your project. If anything is unclear in the following criteria then please do get in contact.

Final Project Mark

The spreadsheet shown below will be used to record:

- The assessment weighting distribution you selected for your project
- The level of attainment demonstrated against the category of assessment (this is for your final project submission)
- The overall project mark
- The adjusted peer weighting given to each member of the team and their individual overall mark (based on the weighting and overall project mark)



```
==============================================================
                                          Weighting     Assessment
Professionalism                               10
Quality of Architectural Design               15
Use of Graphics/Sound
Extent of Game Features
Complexity of Game Algorithms
Coding Style and Code Quality                 25

          Overall Project Score (/100)              0.0

Peer Assessment (Team Only)              Weighting      Awarded Score
                       [Team member]
                       [Team member]
                       [Team member]
                       [Team member]


==============================================================
```

Assessment Category: **Professionalism**
Category Weighting: **10 marks**

Professionalism Overview:

To what extent have you gone about creating your project in a professional manner? In the context of this module, professionalism can be decomposed into two related notions:

- Professionalism during the development of the project. Did you approach the development of the project in a professional manner? Working consistently throughout the semester; making good use of code versioning and spreading development activity across the semester.

- Professional quality of the developed project. Have you developed a high quality game (in terms of a consistent visual appearance, high levels of stability (i.e. few bugs), etc.) alongside providing a high quality demonstration (professionally presented)?

Broadly, whilst the quantity of code developed within the project is of importance, it is better to implement a feature with reduced functionality but high quality then to implement a feature with more functionality if that functionality has bugs or only partially works.

1st (70%+) – High levels of professionalism have been demonstrated throughout project development in terms of approach. In addition, high levels of professionalism have been demonstrated in the developed game and associated demonstration.

2.1 (60-70%) – Mostly high levels of professionalism have been demonstrated throughout in terms of approach. In addition, mostly high levels of professionalism have been demonstrated in the developed game and associated demonstration.

2.2 (50-60%) - Acceptable levels of professionalism have been demonstrated throughout in terms of approach. In addition, acceptable levels of professionalism have been demonstrated in the developed game and associated demonstration.

3rd (40-50%) – Mostly acceptable levels of professionalism have been demonstrated in terms of approach. In addition, mostly acceptable levels of professionalism have been demonstrated in the developed game and associated demonstration.

Fail (<40%) - Unacceptable levels of professionalism have been demonstrated throughout in terms of approach. In addition, unacceptable levels of professionalism have been demonstrated in the developed game and associated demonstration.

1st	High	□
	Medium	□
	Low	□
2.1	High	□
	Medium	□
	Low	□
2.2	High	□
	Medium	□
	Low	□
3rd	High	□
	Medium	□
	Low	□
Fail	High	□
	Medium	□
		□

Assessment Category:	**Quality of Architectural Design**
Category Weighting:	**15 marks**

Assessment Overview: To what extent have you designed the architecture of your game (in terms of objects, classes, interfaces, etc.) in a manner that is appropriate to your proposed game? In order to score highly in this section you will need to ensure that your developed architecture is both extensible and reusable.

More specifically, in terms of your submission does it

- Employ an appropriate architectural model for the type of developed game. In particular, does the architecture offer appropriate game structures, e.g. are objects appropriate and fit-for-purpose, have inheritance relationships been used if applicable, is object composition appropriate and do objects interact in an efficient manner (i.e. within game update loops)?
- Facilitate maintenance. In other words, is architectural maintainability enhanced through good encapsulation of algorithmic behaviour within objects and through clean and intuitive forms of interaction between objects.
- Facilitate extensibility through an architectural design that enables new/extended forms of gameplay to be readily added (e.g. new/extended opponents, objectives, A.I., etc.).

1st (70%+) - The game architecture is fully appropriate to the type of game developed. The architectural design enhances both maintainability and extensibility.

2.1 (60-70%) - The game architecture is fully appropriate to the type of game developed. The architectural design enhances maintainability and may also enhancement extensibility.

2.2 (50-60%) - The game architecture is appropriate to the type of game developed, although it may contain a number of minor limitations. The architecture provides some/limited enhancement of maintainability and little/no enhancement of extensibility.

3rd (40-50%) - The game architecture is mostly appropriate to the type of game developed, although it has a number of limitations which introduce unnecessary complications into the game development. The architecture provides little or no enhancement of maintainability and/or extensibility.

Fail (<40%) - The game architecture is not appropriate to the type of game developed and has one, or more, major limitations that severely limit and/or compromise the game development. Little or no consideration of maintainability and/or extensibility has been demonstrated within the design.

1st	High	□
	Medium	□
	Low	□
2.1	High	□
	Medium	□
	Low	□
2.2	High	□
	Medium	□
	Low	□
3rd	High	□
	Medium	□
	Low	□
Fail	High	□
	Medium	□

Project Assessment Criteria

Assessment Category: **Use of Input, Graphics and/or Sound**
Category Weighting: **Project specific**

Assessment Overview: To what extent have you used input, graphics and sound to good effect within your game? Marks will be equally awarded based on the appropriateness and impressiveness of input/graphics/sound within your game. In other words, the use of input/graphics/sound should support and enhance the type of game you have developed The input/graphics/sound should also provide your game with a 'wow' factor – this might include particularly impressive guesture recognition, animations, graphical layering, special effects, context specific sounds, fades, etc. More specifically, in terms of your submission does it

Provide appropriate input/graphics/sounds that:

- Tie in with the type of game developed, i.e. matching the intended game mood, style of play, means of user interaction, etc.
- Communicate their intended function to the player.
- Employ a consistent 'look' and 'feel' throughout the game.

Provide impressive input/graphics/sound. In other words, does the game use:

- High quality graphics/sound and user input in a consistent and supportive manner that enhances the interaction and visual and aural impact of the game.

1st (70%+) - The game input/graphics/sound are fully appropriate to the type of game and clearly communicate their intended function to the player. The input/graphics/sound will provide the game with a clear and consistent 'look' and 'feel' and be impressive in some aspects.

2.1 (60-70%) -.The game input/graphics/sound are appropriate to the type of developed game and clearly communicate their intended function to the player. The input/graphics/sound will either provide the game with a clear and consistent 'look' and 'feel' and/or be impressive in one or more aspects.

2.2 (50-60%) - The input/game graphics/sound are appropriate to the type of game and communicate their intended function. The input/graphics/sound may be consistently applied throughout the game, or, there may be some examples of impressive usage.

3rd (40-50%) - The input/graphics/sound are appropriate to the type of game. The input/graphics/sound may not be consistent throughout the game but communicate their intended function to the player. Little or no highly impressive aspects are included.

Fail (<40%) - The input/game graphics/sound are either not appropriate to the type of developed game or fail to communicate their intended function to the player. The input/graphics/sound may not be consistent. No impressive aspects are included.

1st	High	□
	Medium	□
	Low	□
2.1	High	□
	Medium	□
	Low	□
2.2	High	□
	Medium	□
	Low	□
3rd	High	□
	Medium	□
	Low	□
Fail	High	□
	Medium	□
		□

Assessment Category: **Extent of Game Features**
Category Weighting: **Project specific**

Assessment Overview: To what extent have you included lots of features within your game? Features, in this context, will encompass both the core features of your game and any bells or whistles you might add along the way. For example, you might have introduced a particularly impressive overlay interface to the game, a nice high-score table with persistent top-scores, good in-game help, etc.

Any type of game feature will contribute to the overall mark awarded for this particular category, although in terms of arriving at an overall assessment equal consideration will be given to the quantity and quality of game feature and the extent to which each game feature enhances the overall game. In order to obtain a passing mark in this component it is important that 'core' game features, i.e. those integral to the type of game, are implemented. If it is not possible to implement a core feature then the reasons why should be included.

Important: Appropriate consideration will be given to the size and complexity of the core functionality within the game, i.e. a game which has complex and/or extensive core functionality can score highly without needing much in the way of additional features.

1st (70%+) - Sufficient game features have been implemented to enable the core features of the game to be fully realised. A sizeable number of additional game enhancements may have been implemented, many of which will enhance the overall game.

2.1 (60-70%) -. Sufficient game features have been implemented to enable the core features of the game to be fully realised. A number of additional game enhancements have been implemented, some of which enhance the overall game. In other words, a functional game has been developed within good enhancement beyond the core functionality.

2.2 (50-60%) - Sufficient game features have been implemented to enable the core features of the game to be fully realised. Some/limited additional game enhancements have been implemented. In other words, a functional game has been developed with limited enhancement beyond the modest core functionality.

3rd (40-50%) - Sufficient game features have been implemented to enable the modest core features of the game to be mostly realised. Limited or no game enhancements have been implemented.

Fail (<40%) - Insufficient core game features have been implemented to enable the limited core functionality of the game to be realised. Limited or no game enhancements have been implemented.

1st	High	□
	Medium	□
	Low	□
2.1	High	□
	Medium	□
	Low	□
2.2	High	□
	Medium	□
	Low	□
3rd	High	□
	Medium	□
	Low	□
Fail		□
	High	□
	Medium	□

Project Assessment Criteria

Assessment Category: **Complexity of Game Algorithm**
Category Weighting: **Project specific**

Assessment Overview: To what extent have you developed complex algorithms within your game? For example, marks will be awarded for any particularly impressive artificial intelligence, collision detection, path finding, etc. routines that you have developed. Likewise, marks will be awarded to mathematically rich algorithms, e.g. those involving the use of path projections, etc. Simply making use of a readily available, complex, off-the-shelf algorithm will not attract marks unless you have heavily modified the algorithm within your game.

Any type of algorithm will contribute to the overall mark awarded for this particular category, although in terms of arriving at an overall assessment greater consideration will be given to algorithms that are based upon mathematical manipulation and/or enhance the overall game. Consideration will be given to the number and complexity of algorithm employed within your game. Additionally, consideration will also be given to any form of adapted or modified 'off-the-shelf' algorithm.

1st (70%+) - The algorithms employed within the game are appropriate and permit the core game functionality to function as intended. A good number of the algorithms employed within the game are complex and/or mathematically rich.

2.1 (60-70%) -. The algorithms employed within the game are appropriate and permit the core game functionality to function as intended. Some of the algorithms employed within the game are complex and/or mathematically rich.

2.2 (50-60%) - The algorithms employed within the game are appropriate and permit the core game functionality to function as intended. Little / no algorithms employed within the game are complex and/or mathematically rich.

3rd (40-50%) - The algorithms employed within the game are mostly appropriate and permit most aspects of core game functionality to function as intended.

Fail (<40%) - The algorithms employed within the game are either not appropriate or do not function as intended in respect of enabling core game functionality.

1st	High	□
	Medium	□
	Low	□
2.1	High	□
	Medium	□
	Low	□
2.2	High	□
	Medium	□
	Low	□
3rd	High	□
	Medium	□
	Low	□
Fail	High	□
	Medium	□
		□

Assessment Category: **Coding Style and Code Quality**
Category Weighting: **25 marks**

Assessment Overview: To what extent is your code well structured, easily readable and understandable, and also readily maintainable and extensible? Additionally, to what extent does your code appropriately exploit Java/Android capabilities and available libraries.

More specifically, in terms of your submission does it provide:

- Readable and readily understandable code in terms of variable naming, looping mechanisms, complexity of expression, etc.
- Maintainable and extensible code in terms of structural complexity, co- and inter-dependence between methods, etc.
- Appropriate layout and code documentation in terms of comments, method headers, class headers, etc.
- Effective and appropriate use of the available libraries

1st (70%+) - The coding style is easily readable and readily understood, with an appropriate layout and high quality code documentation. The code is both readily maintainable and extensible and makes effective use of available libraries.

2.1 (60-70%) -. The coding style is easily readable and readily understood, with an appropriate layout and acceptable code documentation. The code may lend itself to ready maintenance and extension and will make effective use of the available libraries.

2.2 (50-60%) - The coding style is acceptable in terms of its readability, understandability, layout and code documentation. The code may not lend itself to ready maintenance and/or extension but it will make appropriate use of available libraries.

3rd (40-50%) - The coding style is mostly acceptable in terms of its readability, understandability, layout and code documentation, with some areas of weakness. The code may be slightly difficult to maintain and/or extend and it may like use of the available libraries.

Fail (<40%) - The coding style may be unacceptable in terms of its readability, understandability, layout and code documentation, with areas of notable weakness. The code is likely to be difficult to maintain and/or extend and it may not make best use of the available libraries.

1st	High	□
	Medium	□
	Low	□
2.1	High	□
	Medium	□
	Low	□
2.2	High	□
	Medium	□
	Low	□
3rd	High	□
	Medium	□
	Low	□
Fail	High	□
	Medium	□